Using Tabu Search to Avoid Concave Obstacles for Source Location

Junqi Zhang, *Senior Member, IEEE*, Huan Liu, Peng Zu, Mengshi Zhao, Cheng Wang, *Senior Member, IEEE*, Aiiad Albeshri, Abdullah Abusorrah, *Senior Member, IEEE*, and MengChu Zhou, *Fellow, IEEE*

***Abstract*— Recently, using a particle swarm optimizer (PSO) to guide robots in a source location problem has attracted widespread interest. While being navigated by PSO, robots are easily trapped into U-shape-like concave obstacles such that they move back and forth cyclically and fail to locate a correct source. Existing obstacle avoidance strategies perform well when robots have information about all obstacles. Yet in many real scenes, robots have no prior information. This work proposes a novel PSO based on Tabu Search (PSO-TS) for robots to locate multiple sources. Instead of traditionally setting obstacles as tabu objects, PSO-TS innovatively sets trapping areas as tabu objects such that robots do not need prior knowledge or expensive hardware and much time to obtain obstacle information. The weighted average velocity of a robot is employed to determine if it is stuck inside an obstacle-induced area. If so, a rectangular tabu area is set to push robots out of the area and prevents robots from searching the same area again. The proposed method can be embedded into various source location algorithms to improve their performance. Its obstacle avoidance capability is proved. Finally, experimental results show the algorithmic compatibility, environmental adaptability and obstacle avoidance performance of the proposed method.**

***Index Terms*— Source location, obstacle avoidance, particle swarm optimizer, tabu search.**

Manuscript received 18 October 2021; revised 26 July 2022, 9 November 2022, and 24 April 2023; accepted 8 June 2023. Date of publication 6 July 2023; date of current version 1 November 2023. This work was supported in part by the Innovation Program of Shanghai Municipal Education Commission under Grant 202101070007E00098; the Shanghai Industrial Collaborative Science and Technology Innovation Project under Grant 2021-cyxt2-kj10; the Shanghai Municipal Science and Technology Major Project under Grant 2021SHZDZX0100; the Fundamental Research Funds for the Central Universities; the National Natural Science Foundation of China under Grant 51775385, Grant 61703279, Grant 62073244, and Grant 61876218; Institutional Fund Projects under Grant IFPIP: 1579-611-1442; FDCT (Fundo para o Desenvolvimento das Ciencias e da Tecnologia) under Grant 0047/2021/A1; and the Shanghai Innovation Action Plan under Grant 20511100500. The Associate Editor for this article was H. Park. *(Corresponding author: MengChu Zhou.)*

Junqi Zhang, Huan Liu, Peng Zu, Mengshi Zhao, and Cheng Wang are with the Department of Computer Science and Technology, Key Laboratory of Embedded System and Service Computing, Ministry of Education, Shanghai Electronic Transactions and Information Service Collaborative Innovation Center, Tongji University, Shanghai 200092, China (e-mail: zhangjunqi@tongji.edu.cn; liuhuan1912@tongji.edu.cn; 1754027@tongji.edu.cn; zmsxsl@tongji.edu.cn; cwang@tongji.edu.cn).

Aiiad Albeshri is with the Department of Computer Science, King Abdulaziz University, Jeddah 21481, Saudi Arabia (e-mail: aaalbeshri@kau.edu.sa).

Abdullah Abusorrah is with CARE Energy Research and Innovation Center, King Abdulaziz University, Jeddah 21481, Saudi Arabia (e-mail: aabusorrah@kau.edu.sa).

MengChu Zhou is with the Macao Institute of Systems Engineering, Macau University of Science and Technology, Macao 999078, China, and also with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA (e-mail: zhou@njit.edu).

Digital Object Identifier 10.1109/TITS.2023.3286973

I. Introduction

THE source location problem has attracted much concern for its various potential applications. In general, the problem requires intelligent agents such as robots and drones to search for sources in unknown environments with certain constraints [1]. In a source location problem, agents are tasked with finding the location that minimizes or maximizes the scalar field. The field can represent environmental characteristics, such as chemical concentrations, light intensities, or heat. This problem exists in various scenarios, such as environment monitoring, search and rescue operations in a dangerous building, chemical spill investigation and automatic navigation [2], [3], [4], [5], [6], [7]. This work focuses on the following situation: robots are able to obtain the signal strength by their on-board sensor and thus locate signal sources. Signal strength reaches the maximum value where a source is located. Since some tasks might arise in a burning building or cave, source location methods are worth considering especially in dangerous environments where humans may not be safe and general methods like GPS cannot work or incur high cost.

A series of studies have been conducted on source location problems, including gradient ascent/descent [8], [9], [10], [11], extremum seeking [12], [13], map construction [2], [14] and bionic behavioral [15], [16], [17], [18], [19] ones. The gradient-based methods drive robots to move along the gradient-ascending/descending direction by tracking the strength of a source signal. Such methods are prone to be trapped in a local minimum. In extremum seeking methods, a single robot is used to collect measurements at different locations which is time-consuming and demands costly maneuvers. The map construction methods combine the approximation theory and estimation criteria to detect information sources. They need numerous sampling sets, which may result in communication delays and excessive power consumption for signal transformation. Bionic behavior-based algorithms are heuristic ones inspired by the behavior of biological swarms such as fish and birds. Particle swarm optimizer (PSO) is one of them. Since PSO is simple to implement, compatible with swarm robots and does not require any prior knowledge of the spatial distribution of source signal, it has been applied to some source location tasks involving swarm robots.

In a PSO-based source location method, robots have stochastic paths and change their next positions every iteration. They are easily trapped by U-shape-like concave obstacles

1558-0016 © 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: TONGJI UNIVERSITY. Downloaded on January 27,2026 at 09:58:38 UTC from IEEE Xplore. Restrictions apply.

such that they move back and forth cyclically in such trapped areas and fail to locate a correct source. Existing obstacle avoidance methods are mainly based on such methods as fuzzy logic controller, artificial potential field, tabu search and control rules. They perform well in simple convex obstacle environments. Yet, there are various complex concave obstacles like U-shape ones in real scenes. It is difficult for robots to acquire prior information about obstacles. When robots are trapped by concave obstacles, the above methods are unable to drive the robots out of the trapping area shaped by the concave obstacles. In such situations, even if robots are driven away from the obstacles, they may be trapped again when the swarm leads the robot to the obstacles. Thus robots may move cyclically and meaninglessly, failing to find the sources. Concave obstacles are common in actual scenarios such as indoor environments and caves. To enable robots to finish tasks in such situations, robots have to avoid concave obstacles.

As a main contribution, this work proposes a novel PSO based on Tabu Search (PSO-TS) for robots to locate multiple sources. Instead of traditionally setting obstacles as tabu objects, we set trapping areas as tabu ones. We propose to use the weighted average velocity of a robot to tell if it is stuck as caused by obstacles. If so, we set a rectangular tabu area covering the position of the robot, thereby pushing robots out of the area and preventing them from moving into the area. Thus robot swarm shares information and cooperates to get out of the trapped areas. We employ an R-tree to handle the storage structure of tabu areas and speed up a search process. In this strategy, robots need no prior information of unknown environments which makes the method adaptable to complex unknown environments.

The rest of the paper is organized as follows. Section II reviews the related work on obstacle avoidance methods. Section III introduces the background of PSO and Tabu search. The principles of the proposed method are given in Section IV. Section V presents the simulation and comparison results. Finally, Section VI concludes this paper.

II. Related Work

Fuzzy logic control is among the earliest methods for obstacle avoidance of robots. Bao et al. [20] propose a fuzzy logic-based obstacle avoidance strategy, which is executed when there are some obstacles at the front and on the sides of a robot. An obstacle avoidance fuzzy controller receives front obstacle distance and side obstacle distance as the input values and outputs the turning angle and velocity values. Majura et al. [21] propose a preference-based behavior control system for a heading control activity. The heading angle is fuzzified into five turning alternatives to guide a robot to avoid obstacles. To generate efficient navigation rules automatically without initial settings of rules by experts, Motlagh et al. [22] propose a neural network and reinforcement learning-based technique to enable a mobile robot to learn expert rules on its own to avoid obstacles.

Artificial potential fields (APF) are also adopted to avoid obstacles [23], [24]. When robots are near obstacles, a repulsive force generated by the potential field drives them away to avoid crashes during searching. Wang et al. [25] propose a source location method combining PSO and APF. APF is used to construct the corresponding repulsive force field to avoid obstacles and collisions. Li and Wu [23] propose a quantum-leading-following-based optimizer to guide robots. It exerts an attractive force on robots to avoid obstacles in an unknown environment. Tang et al. [24] propose a mechanical PSO combined with APFs, which enables robots to locate both single and multi-targets.

TABLE I
Comparison of Obstacle Avoidance Algorithms

Algorithm	Reference	Need obstacle features	Avoid concave obstacles
Fuzzy logic controller	[20]	N	N
	[21]	N	N
	[22]	N	N
Artificial potential field	[23]	Y	Y
	[24]	N	N
	[25]	N	N
Control rule	[1]	N	Y
	[26]	N	N
Tabu search	[27]	Y	Y
	[28]	Y	Y
	[29]	Y	Y
	Ours	N	Y

Note: Y is yes; N is no.

Some control rules are proposed to avoid obstacles. Zou et al. [1] propose a circumnavigation strategy to avoid obstacles. Once a robot meets an obstacle, it switches to obstacle avoidance mode and starts to circumnavigate the obstacle. As it circumnavigates an area, it measures signal strength along its path. After circumnavigating the entire obstacle, the robot finds a position with the largest signal strength and switches to a source seeking mode. Tang et al. [26] propose a rotation strategy to bypass obstacles, which is effective when the shape of obstacles is rectangular or circular but not concave.

Tabu search is considered as an effective method to avoid obstacles in [27], [28], [29], and [30]. Masehian and Amin-Naseri [27] apply a tabu search to the path planning of mobile robots for the first time. This method works by storing observable angles of obstacles as search neighborhoods in a tabu table. Then a tabu search is used to determine the moving direction of a robot. Zhang et al. [28] combine a tabu search algorithm and a fuzzy logic control strategy to avoid obstacles. It improves the adaptability of mobile robots in changing environments. In such algorithms [29], [30], Robots first detect and search the edges of obstacles, and add the angle points of obstacles to a tabu list. Then, robots visit those positions in the tabu list to check if a source is achievable from there and finally bypass the obstacles. This method is adopted in path planning and assumes that a robot has the ability to detect the geometric features of obstacles, which may not be implementable in some situations such as rough terrain or dark cave and poses a high demand for sensing abilities of robots. The above obstacle avoidance algorithms are compared in Table I. Fuzzy logic control-based algorithms do not rely on prior obstacle features, but cannot avoid concave obstacles. In APF-based algorithms, robots require the radius of a concave obstacle in advance to avoid it. Tabu search-based

Authorized licensed use limited to: TONGJI UNIVERSITY. Downloaded on January 27,2026 at 09:58:38 UTC from IEEE Xplore. Restrictions apply.

TABLE II
NOTATIONS IN PSO-TS

$\boldsymbol{x}_i$=(x_i, y_i)	position of the ith particle(robot)
$\boldsymbol{v}_i$	velocity of the ith particle(robot)
$\boldsymbol{p}_i$	personal best position of the ith particle(robot)
$\boldsymbol{g}$	best position founded by the swarm
t	current iteration step
c_1 and c_2	accelerate coefficients
ϵ_1 and ϵ_2	two random numbers within $[0, 1]$
ω	inertia weight
$\widehat{V}$	upper limit of the robot's moving distance in one time slice
X	a set of feasible positions in the search space
$\mathbb{L}$	a tabu list
θ	a saved object in an R-tree
Ξ	hyperplane rectangle bounding the object θ
Z	a lower node in an R-tree
$\mathbb{N}(\boldsymbol{x})$	a set of feasible positions for the next move of a robot $\boldsymbol{x}$
M and m	the maximal and minimal number of entries in a node of an R-tree
H	a rectangle in an R-tree
$U(\boldsymbol{x}_i)$	open space of a concave obstacle for a robot $\boldsymbol{x}_i$
$\bar{U}(\boldsymbol{x}_i)$	tabu space of a concave obstacle for a robot $\boldsymbol{x}_i$
$\boldsymbol{c}$	the centroid of a tabu rectangle
$\bar{\boldsymbol{v}}_i^t$	the weighted average velocity
α	weight decay coefficient
γ	a threshold to measure the weighted average velocity
N	the number of robots
Ω	a source
T	the maximum number of iterations
s	success rate
d_s	a threshold to measure if a robot converges to a source
d_c^t	convergence distance of the swarm in the kth iteration
N_c	the number of robots whose distance to a source is less than d_s

algorithms require obstacles' inflection point information to avoid them. In control rule-based algorithms, the circumnavigation strategy is an efficient method to avoid obstacles without prior knowledge of obstacles. But, it needs to spend much time navigating around obstacles.

III. BACKGROUND

This section gives the notations in Table II. Then, a basic PSO with physical constraints, a tabu search algorithm and an R-tree structure are introduced.

A. PSO With Physical Constraints

PSO is a stochastic global optimization technique inspired by bird flocking or fish schooling [31], [32]. It has become one of the most popular optimization techniques and has been successfully applied to many fields [33], [34], [35], [36]. In the original PSO, a group of random particles (solutions) is initialized. Each particle in a swarm updates itself by tracking both its personal best position located by itself and the overall best position found so far by the whole swarm in each iteration. To balance global exploration and local exploitation better, an inertia weight parameter ω was introduced to adjust the influence of the previous particle velocities during the optimization process [37]. In this work, a particle uses the following formula to update its velocity and position:

$$v_{ij}^t = \omega v_{ij}^{t-1} + c_1\epsilon_1(p_{ij}^{t-1} - x_{ij}^{t-1}) + c_2\epsilon_2(g_j^{t-1} - x_{ij}^{t-1}), \quad (1)$$

$$x_{ij}^t = x_{ij}^{t-1} + v_{ij}^{t-1} \quad (2)$$

where D is the dimension of the solution space, $j \in \{1, \ldots, D\}$. $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$ refers to the position of the ith particle; $\boldsymbol{v}_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$ is the velocity of the ith particle; $\boldsymbol{p}_i = (p_{i1}, p_{i2}, \ldots, p_{iD})$ is the personal best position of the ith particle; $\boldsymbol{g} = (g_1, g_2, \ldots, g_D)$ is the best position founded by the swarm; t represents the current iteration step; c_1 and c_2 are accelerate coefficients determining the relative importance of $\boldsymbol{p}_i$ and $\boldsymbol{g}$; and ϵ_1 and ϵ_2 are two random numbers within $[0, 1]$. By adjusting the value of ω, particles have a greater tendency to eventually locate the area containing the best fitness value and explore that area in detail. It is suggested to make ω as a dynamic decreasing value, from a value greater than 1.0 to encourage early exploration at the beginning to a value less than 1.0 for searching in details eventually.

In each iteration, for a maximization problem, the vectors $\boldsymbol{p}_i$ and $\boldsymbol{g}$ are updated by:

$$\boldsymbol{p}_i^t = \begin{cases} \boldsymbol{p}_i^{t-1}, f(\boldsymbol{x}_i^t) \le f(\boldsymbol{p}_i^{t-1}) \\ \boldsymbol{x}_i^t, f(\boldsymbol{x}_i^t) > f(\boldsymbol{p}_i^{t-1}) \end{cases}, \quad (3)$$

$$\begin{cases} f(\boldsymbol{g}^t) = max(f(\boldsymbol{p}_1^t), f(\boldsymbol{p}_2^t) \ldots f(\boldsymbol{p}_N^t)) \\ \boldsymbol{g}^t \in \{\boldsymbol{p}_1^t, \boldsymbol{p}_2^t \ldots \boldsymbol{p}_N^t\} \end{cases}. \quad (4)$$

To make the algorithm feasible in real environment, a parameter $\widehat{V}$ is often used to limit each particle's velocity component within $[-\widehat{V}, \widehat{V}]$. If the velocity is not in the interval, it will be modified by:

$$v_{ij} = \widehat{V}\frac{v_{ij}}{|\boldsymbol{v}_i|} \quad (5)$$

B. Tabu Search Algorithm

Tabu Search (TS) is a metaheuristic method that guides a local heuristic search procedure to explore a solution space beyond local optimality [38]. One advantage of the Tabu search technique is its ability to avoid search and evaluation of the visited space by using a flexible memory of the search history. TS algorithm has been applied successfully for various classic and practical problems such as scheduling [39] and character recognition [40].

Denote the searching space as set X, the *move* action is defined to transform solution $\boldsymbol{x}^t \in X$ into another solution $\boldsymbol{x}^{t+1} \in X$, which can be marked as $\boldsymbol{x}^{t+1} = move(\boldsymbol{x}^t)$. For any solution $\boldsymbol{x} \in X$, the neighborhood set of $\boldsymbol{x}$ is generated by all the feasible *move* of $\boldsymbol{x}$, marked as $\mathbb{N}(\boldsymbol{x}) = \{\boldsymbol{x}' | \boldsymbol{x}' = move(\boldsymbol{x}), \boldsymbol{x}' \in X\}$. Tabu List ($\mathbb{L}$) is adopted to record the solutions that have been searched to avoid the repeated search and endless loop. $\mathbb{L}$ is defined as a list with a certain length. Any solution in $\mathbb{L}$ will not be searched. Stopping criteria of the algorithm includes reaching the maximum number of iteration steps, obtaining a satisfactory solution, etc. The algorithm is given in Algorithm 1.

C. R-Tree Structure

R-tree is a data structure obtained by extending B-tree [41] to multi-dimensional situations [42]. R-tree is a height-balanced tree with indexed records in its leaf nodes. The elements saved in the tree are spacial objects, and records

Authorized licensed use limited to: TONGJI UNIVERSITY. Downloaded on January 27,2026 at 09:58:38 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1 Tabu Search Algorithm

Input: Initial solution $\boldsymbol{x}^0$, fitness function $f(\boldsymbol{x})$.
Output: Minimal solution $\boldsymbol{x}^*$.

```
Initialize x* = x^0, x = x^0, L = ∅.
while Not satisfying the stopping criteria do
    N̄ = N(x) − L.
    x̂' = arg min_{x'∈N̄} f(x').
    if x̂' ≠ ∅ then
        if f(x̂') < f(x*) then
            x* = x̂'.
        end if
        Push x to L.
        x = x̂'.
    end if
end while
return x*.
```

contained in the leaf nodes of the R-tree are in the following form:

$$(\Xi, \theta)$$

where θ refers to a saved object and Ξ is a hyperplane rectangle bounding the object with the form

$$\Xi = (\Xi_0, \Xi_1, \ldots, \Xi_{n-1})$$

where n is the dimension and each Ξ_k is a closed interval $[\Xi_{kl}, \Xi_{kr}]$. For the non-leaf node, it contains records in the form

$$(\Xi, Z)$$

where Z points to a lower node in the tree and Ξ is a rectangle covering all sub-rectangles of its child nodes.

Denoting the maximal and minimal number of entries in one node as M and m, then an R-tree satisfies the following properties:

1) $m \leq \frac{M}{2}$.
2) In each entry (Ξ, Z) of a non-leaf node, I is the smallest rectangle covering all rectangles in its child node.
3) Except the root, every leaf node contains records between m and M.
4) Except the root, every non-leaf node contains children between m and M.
5) If the root is not a leaf, it has at least two children.
6) All leaf nodes are in the same level.

For searching whether a rectangle H is covered by an R-tree, the algorithm checks each entry to determine whether H is covered by a rectangle saved in the entry. For all entries satisfying the condition, searching is invoked in their child subtrees. Moreover, when S degenerates into a point, it means checking whether a point is covered by an R-tree. For an insertion, R-tree locates a feasible place by minimizing an enlarged area when adding a new entry. Once completing the insertion, it adjusts its all nodes to satisfy the above properties. Compared with other structures, an R-tree is especially suitable for spatial data storage and for tabu areas storage. In each iteration, a robot needs to determine whether its next position is in a tabu area or not. The advantage of an R-tree is its ability to quickly search for an area. According to [41], the time complexity of an R-tree to search for a tabu area is only $O(log(n))$.

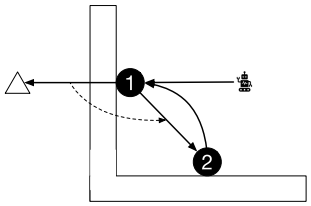


Fig. 1. A robot is trapped by the obstacle. It repeats moving between points 1 and 2.

IV. PSO With Tabu Search

A. Definition of Deadlocks for Obstacle Avoidance

Most obstacle avoidance methods have good performance under the premise that obstacles in unknown environments are convex or known. If concave obstacles appear in an environment or the prior knowledge of obstacles is unknown, such as L-shaped-like and U-shaped-like obstacles, a robot may be trapped and cannot reach sources. Consider the situation in Fig. 1. The source is denoted as a symbol '△' and is on the left side of an L-shaped-like obstacle. The robot is at point 1 on the right side of the obstacle. The robot regards the source as its global best and keeps rotating counterclockwise when finding obstacles in its detecting range. After several rotations, the robot reached point 2. In the next iteration, according to PSO, the robot returns to the neighborhood of point 1. Hence it moves cyclically between points 1 and 2 and is trapped by this obstacle. Such a phenomenon is defined as a deadlock in this work. The proposed algorithm attempts to solve such a deadlock problem with as little additional cost as possible.

B. The Proposed Tabu Search-Based PSO (PSO-TS)

When a robot is trapped by an obstacle, it may make repeated and meaningless movements in a certain area. Denote the velocity of a robot at tth iteration as $\boldsymbol{v}_i^t$, during several iterations, the distance between the starting and the final positions of the robot $||\Delta \boldsymbol{x}||$ is very small, which means $\Delta \boldsymbol{x}_i^{p,q} = \Sigma_{t=p}^{q} \boldsymbol{v}_i^t$ will be limited to a certain range.

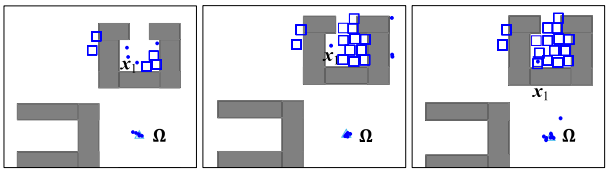
To determine whether a robot is stuck, the weighted average velocity $\hat{\boldsymbol{v}}_i^t$ is employed to learn if it is stuck by an obstacle. The recursion formula of the weighted velocity is as follows:

$$\hat{\boldsymbol{v}}_i^1 = \boldsymbol{v}_i^1, \tag{6}$$

$$\hat{\boldsymbol{v}}_i^{t+1} = \hat{\boldsymbol{v}}_i^t + \alpha(\boldsymbol{v}_i^{t+1} - \hat{\boldsymbol{v}}_i^t) \tag{7}$$

where $\alpha \in [0, 1]$ is the weight decay coefficient. After an iteration, the weighted average velocity is updated according to this formula, and a robot checks whether it is trapped based on it. This weighted average velocity characterizes the motion state of a robot over a period of time and is composed of the velocity in each iteration with different weights. In the weighted average velocity, the proportion of early iterations is smaller, while the one of the recent iteration is larger.

Authorized licensed use limited to: TONGJI UNIVERSITY. Downloaded on January 27,2026 at 09:58:38 UTC from IEEE Xplore. Restrictions apply.



(a) $t = 60$ (b) $t = 90$ (c) $t = 110$

Fig. 2. An example of a robot being trapped by tabu areas.

If the norm of a robot's weighted average velocity in the tth iteration $\hat{\boldsymbol{v}}_i^t$ is less than a threshold γ, it means the position of the robot in a period of time has almost not changed. If the robot does not get closer to the source, it is probably trapped by obstacles. In a real environment, since robots have no idea where the sources are, they regard their global bests as the sources. Thus they evaluate the distance between their personal and global bests. Based on the above analysis, the criteria for judging whether a robot is trapped can be set as follows:

1) The weighted average velocity is less than a threshold, i.e. $\hat{\boldsymbol{v}}_i^t < \gamma$.
2) The positions of personal and global bests keep a certain distance, i.e. $||\boldsymbol{p}_i^t - \boldsymbol{g}^t|| > d$ for a fixed d.

Once the above condition is satisfied, the algorithm pushes tabued rectangle areas to an R-tree. All robots in the tabued rectangle will leave the area outward along the line between themselves and the center of the tabu area. The moving distance is inversely proportional to the distance from the center of the area to its location, and it is limited in $[\widehat{V}/2, \widehat{V}]$:

$$\dot{\boldsymbol{v}}_i^t = \frac{(\boldsymbol{x}_i^t - \boldsymbol{c})(\widehat{V} - |\boldsymbol{x}_i^t - \boldsymbol{c}|)}{|\boldsymbol{x}_i^t - \boldsymbol{c}|} \tag{8}$$

where $\boldsymbol{c}$ is the centroid of a tabu rectangle. If there is any obstacle in this direction, the robot rotates its direction counterclockwise to find a new path. Starting from the next iteration, robots consider the areas in the tabu list as obstacles.

When multiple robots enter a concave obstacle, they simultaneously generate tabu areas to avoid the obstacle as shown in Fig. 2(a). Since each robot generates tabu areas based on its own weighted velocity, it may be surrounded by tabu areas generated by its peers as shown in Fig. 2(b). Then, robot $\boldsymbol{x}_1$ is trapped into tabu areas and the obstacle as shown in Fig. 2(c). In this case, the robot has no way to go. Then, a backtracking mechanism is triggered to reverse the path traveled until the robot exits the tabu areas. The proof of avoiding concave obstacles using the proposed method is detailed in Section IV-C.

To avoid collision, a warning distance is defined as the hypothetical robot's braking distance and is set to 0.1. Then, the warning zone of a robot is a circular area with its position as the center and the warning distance as the radius. When a robot enters the warning zones of other robots, it avoids them by rotating itself [26]. This work takes 15 degrees for each rotation. After several rotations (maximum 23 times in this work), if a robot cannot find a suitable collision-free path, it stays at its current position until the next time slice.

The flow chart of the proposed PSO-TS is shown in Fig. 3. The tabu search strategy is added to the "Movement and obstacle avoidance". It should be noted that embedding the tabu search strategy does not affect the mechanics-based obstacle avoidance strategy of the original algorithm. The existing source location algorithms may have their own obstacle avoidance strategies like the artificial potential field method. After embedding the tabu search strategy, the algorithm first executes the existing obstacle avoidance strategy and then executes our embedded strategy. Thus, the performance of these algorithms will be promoted on the concave obstacle avoidance. As the embedded algorithm only focuses on escaping from deadlock status and does not affect the searching process when robots are not trapped by obstacles, so this tabu strategy has good compatibility and can be easily adopted by various searching strategies. The complete algorithm is given in Algorithm 2.

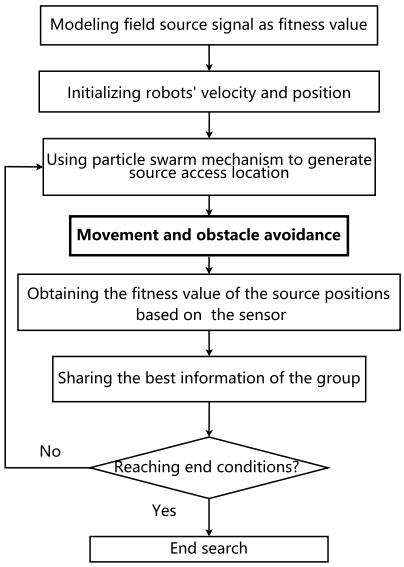


Fig. 3. The flow chart of PSO-TS where the proposed method focuses on the "Movement and obstacle avoidance step".

C. Rationales of Avoiding Concave Obstacles

Considering a single-source location problem in a concave obstacle environment, we prove the obstacle avoidance ability of the proposed tabu search method. Since the proposed method can combine with any source location algorithms to avoid concave obstacles, we do not limit which source location algorithm to use. Therefore, a reasonable assumption is that a robot can converge to a position with the strongest signal using a source location algorithm in an obstacle-free environment. In this work, for a robot $\boldsymbol{x}_i$, a concave obstacle's entire interior space can be divided into two types: open space $U(\boldsymbol{x}_i)$ and tabu one $\bar{U}(\boldsymbol{x}_i)$. Then, there are two states for robot $\boldsymbol{x}_i$ in a concave obstacle's interior space:

State 1: the robot searches in $U(\boldsymbol{x}_i)$.

State 2: the robot is trapped into $\bar{U}(\boldsymbol{x}_i)$.

First, we prove that if a robot is in State 1, using the proposed method, it can fill up the interior open space of a concave obstacle by generating tabu areas and gets out of the concave obstacle in a limited time. Then, we prove that if a robot is in State 2, it is able to switch to State 1 by a

Authorized licensed use limited to: TONGJI UNIVERSITY. Downloaded on January 27,2026 at 09:58:38 UTC from IEEE Xplore. Restrictions apply.

Algorithm 2 PSO-TS

Input: the number of robots N, the maximum number of iterations T, fitness function $f(x)$, accelerate coefficients c_1 and c_2.

Output: The best position $\boldsymbol{g}$ founded by the swarm.

```
Initialize positions of robots, tabu list 𝕃 as an empty R-tree, iteration number t = 1.
while t ≤ T do
    Driving robots searching the source by (1) to (5).
    Updating the weighted velocity of each robot by (6) and (7).
    for i = 1 → N do
        if v̂_i^t < γ then
            Push the position of the robot to 𝕃.
            Push robots out of the tabu areas by (8).
        end if
        if a robot is trapped by tabu areas then
            A backtracking mechanism is triggered.
        end if
    end for
    t = t + 1.
end while
return g.
```

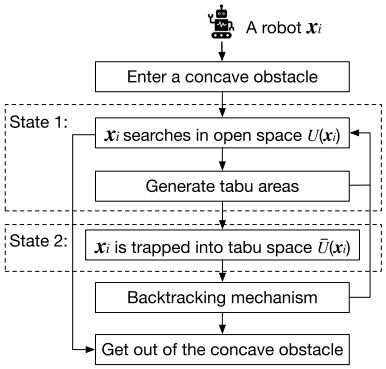


Fig. 4. State switching of a robot in a concave obstacle.

backtracking mechanism. Using the proposed method, a robot switches its state in a concave obstacle as shown in Fig. 4.

Theorem 1: There exists constants $t' < \infty$ and $\varepsilon > 0$, such that $\forall t > t'$, $\forall \gamma > \varepsilon$, a robot can get out of a concave obstacle, where t is the current iteration count, and γ is a threshold to measure the weighted average velocity.

Proof: According to the assumption that a robot can converge to a position with the strongest signal using a source location algorithm, a robot can move towards an attraction point when it meets a concave obstacle. The attraction point is defined as a position with the strongest signal strength within a concave obstacle. There are one or multiple consecutive attraction points inside a concave obstacle as shown in Fig. 5. From Fig. 5(a), when a robot moves to attraction point A, it rotates a certain number of degrees counterclockwise to an oscillation point B in the next iteration. After that, the robot oscillates between A and B. From Fig. 5(b), a robot moves to an arbitrary attraction point. Since the robot always rotates

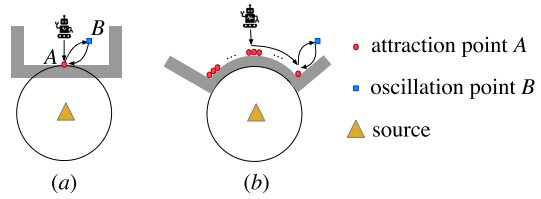


Fig. 5. Cases of attraction points in a concave obstacle.

counterclockwise to avoid collisions, it next moves to the rightmost attraction point and oscillates between the rightmost attraction point and an oscillation point instead of moving back and forth between multiple attraction points. Therefore, if a robot searches in State 1, it causes an oscillation between an attraction point and the oscillation point. A robot's condition for generating tabu areas is analyzed as follows.

According to (7), the expectation of weighted average velocity $\hat{\boldsymbol{v}}_i^t$ in the last k iterations can be expressed as

$$E(|\hat{\boldsymbol{v}}_i^t|) = E(|\Sigma_{\tau=t-k+1}^{t}\alpha(1-\alpha)^{t-\tau}\boldsymbol{v}_i^{\tau}|), \tag{9}$$

where t is the current iteration index. When a robot oscillates between A and B in the last k iterations, we have $0 < |\boldsymbol{v}_i^t| = |A - B| \leq \widehat{V}$. Then, the expectation of $\hat{\boldsymbol{v}}_i^t$ can be deduced as

$$\begin{aligned} E(|\hat{\boldsymbol{v}}_i^t|) &= E(|\Sigma_{\tau=t-k+1}^{t}(-1)^{t-\tau}\alpha(1-\alpha)^{t-\tau}\boldsymbol{v}_i^t|) \\ &= \frac{\alpha(\alpha-1)^{k-1}(1-(\alpha-1)^{-k})}{1-\frac{1}{\alpha-1}}E(|\boldsymbol{v}_i^t|) \\ &\leq \frac{\alpha-\alpha(\alpha-1)^k}{2-\alpha}\widehat{V} \end{aligned} \tag{10}$$

Give a fixed k, when a robot oscillates more than k iterations, its weighted average velocity is less than or equal to a constant $\varepsilon = \frac{\alpha-\alpha(\alpha-1)^k}{2-\alpha}\widehat{V}$. Thus, if we set $\gamma > \varepsilon$ where γ is a threshold to measure the weighted average velocity, according to Algorithm 2, a robot can generate a tabu area as long as it oscillates k iterations.

Since the interior open space of a concave obstacle is limited, a source location algorithm can drive a robot to locate the attraction point in a limited time interval t_0. Then, there exists a time constant $t' = t_0 + k < \infty$ and a constant $\varepsilon = \frac{\alpha-\alpha(\alpha-1)^k}{2-\alpha}\widehat{V}$, such that $\forall t > t'$ and $\forall \gamma > \varepsilon$, we have that the robot can generate tabu areas. Then, the generated tabu areas can fill up the concave obstacle in a limited time. Hence, the robot can be pushed out of a concave obstacle by tabu areas. Therefore, if a robot searches in State 1 and we set $\gamma > \varepsilon$, it can fill up the interior open space of a concave obstacle by generating tabu areas and gets out of the concave obstacle in a limited time.

If a robot is in State 2, a backtracking mechanism is triggered to reverse the path traveled until the robot exits the tabu space. Since a robot starts its search outside the concave obstacle, it must be able to reversely escape from the tabu space and switch to State 1. Once it is at State 1, it repeats till U is empty or exits out of the concave obstacle. In summary, the robot must be able to get out of the concave obstacle by using the proposed method. ■

Authorized licensed use limited to: TONGJI UNIVERSITY. Downloaded on January 27,2026 at 09:58:38 UTC from IEEE Xplore. Restrictions apply.

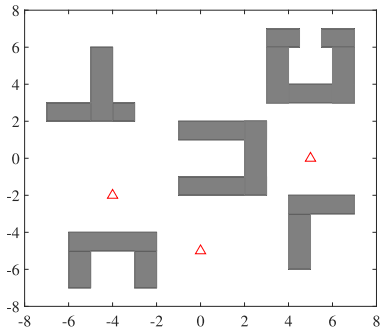


Fig. 6. Scenario 1 contains concave obstacles with right corners.

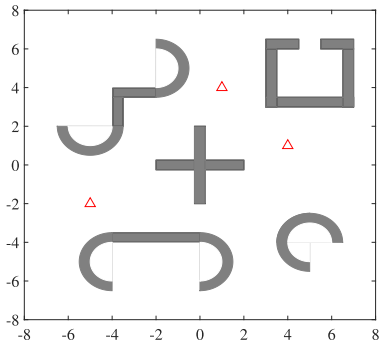


Fig. 7. Scenario 2 contains concave obstacles with rounded corners.

V. Experimental Results

Two source location scenarios are used to run the simulation experiments as shown in Figs. 6 (scenario 1) and 7 (scenario 2). They contain concave obstacles with right and rounded corners, respectively. In scenario 1, two environments corresponding to two distributions of sources are used. Distribution 1: $[(5,0);(0,-5);(-4,-2)]$; Distribution 2: $[(6,1);(0,5);(-4,0)]$. Their corresponding fitness functions (signal strengths) are as follows:

$$f_1^1(\boldsymbol{x}) = 5 \cdot exp\left(-\left\|\frac{\boldsymbol{x}-(5,0)}{5}\right\|\right) + 5 \cdot exp\left(-\left\|\frac{\boldsymbol{x}-(0,-5)}{5}\right\|\right) + 5 \cdot exp\left(-\left\|\frac{\boldsymbol{x}-(-4,-2)}{5}\right\|\right). \quad (11)$$

$$f_1^2(\boldsymbol{x}) = 5 \cdot exp\left(-\left\|\frac{\boldsymbol{x}-(6,1)}{5}\right\|\right) + 5 \cdot exp\left(-\left\|\frac{\boldsymbol{x}-(0,5)}{5}\right\|\right) + 5 \cdot exp\left(-\left\|\frac{\boldsymbol{x}-(-4,0)}{5}\right\|\right). \quad (12)$$

In scenario 2, two environments corresponding to two distributions of sources are used. Distribution 1: $[(4,-2);(-3,-2);(2,2)]$; Distribution 2: $[(4,1);(-5,-2);(1,4)]$. Their corresponding fitness functions (signal strengths) are as follows:

$$f_2^1(\boldsymbol{x}) = 5 \cdot exp\left(-\left\|\frac{\boldsymbol{x}-(4,-2)}{5}\right\|\right) + 5 \cdot exp\left(-\left\|\frac{\boldsymbol{x}-(-3,-2)}{5}\right\|\right) + 5 \cdot exp\left(-\left\|\frac{\boldsymbol{x}-(2,2)}{5}\right\|\right). \quad (13)$$

$$f_2^2(\boldsymbol{x}) = 5 \cdot exp\left(-\left\|\frac{\boldsymbol{x}-(4,1)}{5}\right\|\right) + 5 \cdot exp\left(-\left\|\frac{\boldsymbol{x}-(-5,-2)}{5}\right\|\right) + 5 \cdot exp\left(-\left\|\frac{\boldsymbol{x}-(1,4)}{5}\right\|\right). \quad (14)$$

The proposed tabu search strategy can be embedded in various source location algorithms. To verify the adaptability of the proposed method, four source location algorithms including two PSO-based algorithms and two non-PSO-based algorithms are conducted. They are Searching Auxiliary Points-based PSO (SAPPSO) [26], Multi-source Mechanical PSO (MMPSO) [24], Probabilistic Finite State Machine-based strategy for multi-target search (PFSM) [43] and Improved Group Explosion Strategy for searching multiple targets (IGES) [44]. The parameters of four source location algorithms are shown in Table III. To verify the efficiency of the proposed tabu search strategy, three obstacle avoidance methods are added for comparison, which are Artificial Potential Fields-based (APF) [25], Fuzzy logic-based [20] and control Rule-based [1] methods. All algorithms are carried out by using MATLAB R2016a on a PC with Intel(R) Core(TM) i5 running at 3.10GHz with 4G RAM.

TABLE III
Parameters of Source Location Algorithms

Algorithm	Parameter Setting	Reference
SAPPSO	$c_1 = c_2 = 2.1$, $T = 300$, $\widehat{V} = 0.5$, $\eta = 0.1$, $f_{th} = 4$, $l_d = 6$	[26]
MMPSO	$c_1 = c_2 = 2.1$, $T = 300$, $\widehat{V} = 0.5$, $\eta = 1.05$, $\delta = 1.03$, $n_d = 3$, $r_d = 1$	[24]
PFSM	$P_{ini} = 0.9997$, $w = 0.55$, $T_b = 2.3$, $N = 30$, $T = 300$, $\widehat{V} = 0.5$	[43]
IGES	$\beta_G = 15$, $H = 10$, $N = 30$, $T = 300$, $\widehat{V} = 0.5$	[44]

To determine whether robot i has found a source, its personal best position $\boldsymbol{p}_i$ is checked. If the distance between $\boldsymbol{p}_i$ and a source is less than a threshold d_s=0.5, then the robot succeeds. In this work, the success rate s, convergence distance d_c, and d_c^t are three indicators for comparison. Their formulas are as follows.

$$s = \frac{N_c}{N} \quad (15)$$

where N_c is the number of robots that the distance between their personal best position $\boldsymbol{p}_i$ and the source is less than d_s.

$$d_c^t = \Sigma_{i=1}^{N} |\boldsymbol{p}_i^t - \boldsymbol{s}_i| \quad (16)$$

where $\boldsymbol{s}_i$ is the position of the closest source for robot i. d_c is defined as d_c^t at the end of the algorithm:

$$d_c = \Sigma_{i=1}^{N} |\boldsymbol{p}_i^T - \boldsymbol{s}_i| \quad (17)$$

where T is the maximum number of iterations.

Authorized licensed use limited to: TONGJI UNIVERSITY. Downloaded on January 27,2026 at 09:58:38 UTC from IEEE Xplore. Restrictions apply.

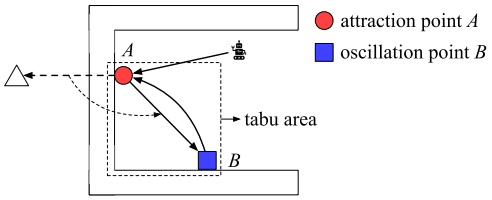


Fig. 8. A robot is trapped by the obstacle. It repeats moving between points A and B.

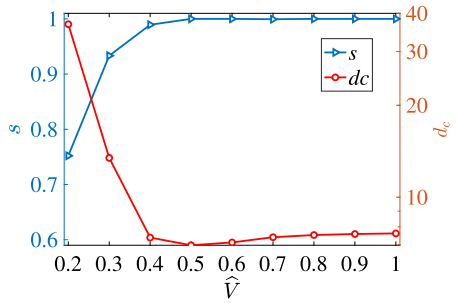


Fig. 9. Success rate (s) and convergence distance (d_c) of SAPPSO-TS with parameter $\widehat{V}$ on different settings.

A. Parameter Analysis

The size of the tabu side length is an important factor in the algorithm's performance. If a tabu area is too large, it may cause some passable areas to be tabued, which increases the movement cost of the robot swam. Conversely, if the area is too small, robots need more time and movement costs to avoid obstacles. Therefore, it is necessary to consider how to select an appropriate size for a tabu area. When a robot meets a concave obstacle as shown in Fig. 8, it can move towards an attraction point A by source location algorithms. The attraction point is defined as a position with the strongest signal strength within a concave obstacle. Then, the robot is blocked by the obstacle. It rotates a certain number of degrees counterclockwise to an oscillation point B in the next iteration. After that, the robot oscillates between A and B and deadlocks. The area around A and B is considered as an impassable area that needs to be tabooed. Since the upper limit of the robot's moving distance in a time slice is $\widehat{V}$, then, $||A - B|| \leq \widehat{V}$. Thus a rectangle centered at the midpoint between points A and B with side length $\widehat{V}/\sqrt{2}$ is set as a tabu area.

In (10), the threshold γ is determined by three parameters $\widehat{V}$, α and k. $\widehat{V}$ is the upper limit of the robot's moving distance in a time slice. Robots with large $\widehat{V}$ have more space for exploration, while they explore space in a fine manner with small $\widehat{V}$. Fig. 9 shows the success rate (s) and convergence distance (d_c) of SAPPSO-TS with $\widehat{V}$ taking different values. From Fig. 9, when $\widehat{V} \in [0.2, 0.5]$, as $\widehat{V}$ increases, s increases and d_c decreases sharply because robots are more explorative. When $\widehat{V} \in [1, 0.5]$, as $\widehat{V}$ increases, s is stable at around 1. d_c slightly increases because the search accuracy of robots gradually decreases. Then, $\widehat{V} = 0.5$ is chosen as an appropriate value. $\alpha \in [0, 1]$ is a weight decay coefficient. From (9), a large value makes robots focus on short-term vision, while a small value makes them focus on long-term vision. According to some trial tests, α is set as a small value 0.2 to give robots a long-term vision. Since early iterations have almost no effect on the weighted average velocity, the last $k = 6$ iterations are considered to calculate the weighted average velocity.

TABLE IV
EXPERIMENTAL RESULTS IN SCENARIO 1

Algorithm	f_1^1		f_1^2	
	s	d_c	s	d_c
SAPPSO-APF	0.789	36.810	0.779	36.848
SAPPSO-Fuzzy	0.797	30.165	0.847	24.335
SAPPSO-Rule	0.963	9.689	0.947	13.038
SAPPSO-TS	**0.993**	**7.872**	**0.989**	**8.178**
MMPSO-APF	0.795	39.127	0.829	35.442
MMPSO-Fuzzy	0.815	28.635	0.838	25.211
MMPSO-Rule	**0.988**	**8.482**	0.987	8.488
MMPSO-TS	0.978	9.169	**0.995**	**7.517**
IGES-APF	0.788	25.637	0.769	26.607
IGES-Fuzzy	0.850	16.582	0.900	18.363
IGES-Rule	0.967	14.403	0.973	13.308
IGES-TS	**0.986**	**12.902**	**0.978**	**12.122**
PFSMS-APF	0.829	23.020	0.897	18.047
PFSMS-Fuzzy	0.898	14.100	0.924	15.168
PFSMS-Rule	0.886	11.212	0.873	12.201
PFSMS-TS	**0.983**	**9.929**	**0.969**	**9.913**

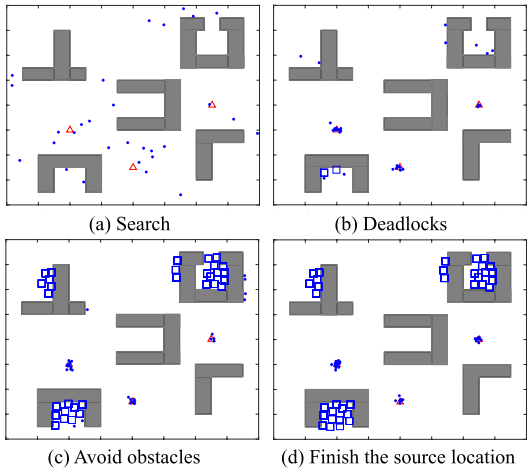


Fig. 10. The process of robot avoiding obstacles in scenario 1.

B. Comparison Experiment

Four obstacle avoidance methods (including the proposed tabu search) are embedded in four source location algorithms. The form of an algorithm combining a source location method and an obstacle avoidance method is expressed as [source location-obstacle avoidance], e.g., SAPPSO-APF. Then, sixteen algorithms are conducted for comparison. Each algorithm runs 50 times and in each run, 300 iterations are performed. Table IV shows the experimental results in scenario 1. From Table IV, the search performance of the proposed TS and Rule-based methods are obviously better than that of APF and Fuzzy-based ones, because they obtain higher success rates and finer convergence distances.

In SAPPSO, MMPSO and IGES-based algorithms, the Rule and TS-based obstacle avoidance methods show comparable performance. In the Rule-based method, when a robot meets an obstacle, it circumnavigates the obstacle and finds the

Authorized licensed use limited to: TONGJI UNIVERSITY. Downloaded on January 27,2026 at 09:58:38 UTC from IEEE Xplore. Restrictions apply.

TABLE V
EXPERIMENTAL RESULTS IN SCENARIO 2

Algorithm	f_2^1		f_2^2	
	s	d_c	s	d_c
SAPPSO-APF	0.611	43.428	0.620	45.116
SAPPSO-Fuzzy	0.659	33.047	0.647	36.116
SAPPSO-Rule	0.987	7.894	0.987	**7.819**
SAPPSO-TS	**0.994**	**7.423**	**0.988**	7.965
MMPSO-APF	0.643	41.067	0.684	37.345
MMPSO-Fuzzy	0.657	34.290	0.657	33.513
MMPSO-Rule	**0.996**	**8.363**	**0.999**	**7.911**
MMPSO-TS	0.965	10.397	0.975	9.240
IGES-APF	0.645	30.214	0.698	28.062
IGES-Fuzzy	0.600	29.355	0.650	26.556
IGES-Rule	0.950	19.630	**0.960**	15.480
IGES-TS	**0.967**	**19.351**	0.975	**12.296**
PFSMS-APF	0.441	41.816	0.616	31.913
PFSMS-Fuzzy	0.433	42.538	0.597	34.849
PFSMS-Rule	0.705	11.830	0.777	11.239
PFSMS-TS	**0.992**	**11.142**	**0.954**	**10.400**

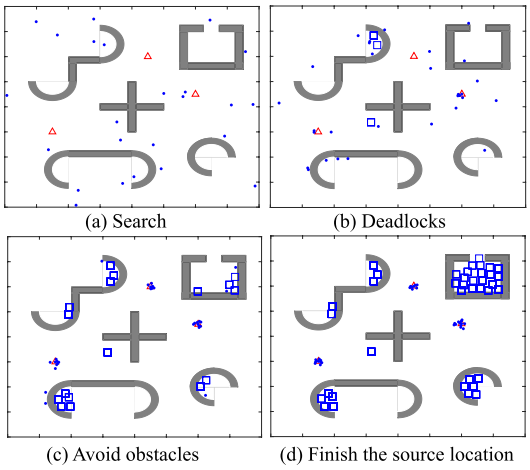


Fig. 11. The process of robot avoiding obstacles in scenario 2.

position with the strongest signal to avoid the obstacle. Thus, the Rule-based method can successfully avoid an obstacle as long as the size of the obstacle is limited. Compared with the Rule-based method, the proposed TS-based method shows excellent obstacle avoidance capability. In four PFSMS-based algorithms, the proposed TS method shows better search performance than the Rule-based one. Since PFSMS consists of three stages including diffusion, search and target processing. The switching of stages causes robots to meet obstacles that have been avoided. Then, the robots with the Rule-based method go the same way, while in the proposed TS, if a concave obstacle is filled up by tabu areas, the robot will not be trapped by the obstacle again. Therefore, the proposed TS has a better capability to adapt to different source search algorithms.

Table V shows the experimental results in scenario 2, which contains more complex concave obstacles with rounded corners. From Table V, the search performance of the proposed TS and Rule-based methods are obviously better than that of APF and Fuzzy-based ones, which is the same as the experimental result of Scenario 1. In four MMPSO-based algorithms,

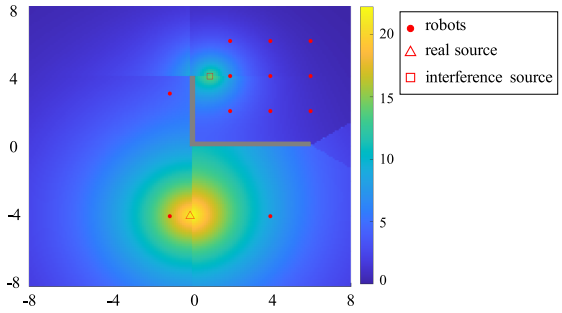


Fig. 12. An example of signal diffusion condition (α=β=0.4) in scenario 3.

the Rule-based method shows better search performance than the proposed TS. Nevertheless, the proposed TS still has a success rate of over 0.96. Comparing the experimental results in Scenarios 1 and 2, all the TS-based algorithms show excellent obstacle avoidance capability with a success rate of over 0.95. This reflects that the proposed TS method works well in different scenarios to avoid concave obstacles. The proposed TS has excellent environmental adaptability.

Figs. 10 and 11 show the process of robots avoiding obstacles in scenarios 1 and 2, respectively. When robots are trapped into concave obstacles, they generate tabu areas to fill up the concave obstacles as shown in Figs. 10(c) and 11(c). Then, in Figs. 10(d) and 11(d), the robots are pushed out of the concave obstacles and finish the source location.

C. Simulation Experiment Considering Obstacle's Physical characteristics

In an actual scene, a signal can be reflected or attenuated by obstacles. When the signal strengths or obstacle materials are different, the degree of signal reflection and penetration differs. To verify the obstacle avoidance ability of the proposed TS in a source location environment with different signal diffusion conditions, scenario 3 is simulated in Fig. 12 that shows an example of signal diffusion condition when α=β=0.4. α and β respectively represent the reflection and penetration ratio of a signal when it encounters an obstacle. For example, α=β=0.4 means that 40% of a signal is refracted, and 40% penetrates the obstacle. The attenuation ratio of a signal is fixed at 20%. Five signal diffusion conditions are simulated including (α, β) = {(0, 0.8); (0.2, 0.6); (0.4, 0.4); (0.6, 0.2); (0.8, 0)}. There is a real source([0, -4]), an interference source ([1], [4]) and 12 robots in scenario 3. The fitness function (signal strengths) consists of two parts, including a real source signal f_3^* and an interference source signal $\tilde{f}_3$, as shown in (18), at the bottom of the next page. In the simulation experiment, four obstacle avoidance algorithms including the proposed TS, APF [25], Fuzzy logic-based [20] and control Rule-based [1] methods are executed by using SAPPSO [26] as the baseline for source location.

In the experiment, initial positions of 12 robots are shown in Fig. 12. Three robots are initialized outside the concave obstacle. For the other nine robots, their path to the real source is blocked by a concave obstacle. Table VI shows the experimental results in scenario 3, which includes the number of trapped robots τ and success rate s. In Table VI, for

Authorized licensed use limited to: TONGJI UNIVERSITY. Downloaded on January 27,2026 at 09:58:38 UTC from IEEE Xplore. Restrictions apply.

SAPPSO-APF, an average of 8.54 robots out of 9 trapped ones get stuck in five signal diffusion conditions. The success rate of converging to the real source is about 29%. For SAPPSO-Fuzzy, about 8.6 robots out of 9 trapped ones get stuck in five signal diffusion conditions. The success rate of converging to the real source is about 28%. It indicates that in five signal diffusion conditions, SAPPSO-APF and SAPPSO-Fuzzy can hardly guide robots to avoid a concave obstacle. For SAPPSO-Rule, when (α, β) = (0.8, 0), 0.2 robots on average get stuck. In the other four signal diffusion conditions where (α, β) = {(0, 0.8); (0.2, 0.6); (0.4, 0.4); (0.6, 0.2)}, about 8.9 robots on average get stuck. In SAPPSO-Rule, if a robot meets an obstacle, it circumnavigates the obstacle and measures signal strength along its path. After circumnavigating the entire obstacle, the robot finds a position with the largest signal strength. The strongest position on the circumnavigating path may be inside the concave obstacle due to the presence of interference sources or due to the signal reflection. In such case, SAPPSO-Rule fails to guide robots to avoid a concave obstacle. For SAPPSO-TS, in five signal diffusion conditions, 9 trapped robots are able to bypass the obstacle and converge to the signal source with a success rate of 1. The obstacle avoidance ability of TS is not affected by obstacle's physical characteristics.

Fig. 13 shows the convergence curves of robots in five signal diffusion conditions. Since SAPPSO-APF and SAPPSO-Fuzzy can hardly guide robots to avoid a concave obstacle, their convergence distances (d_c) maintain a large value (>40). From Figs. 13(a)-(d), in convergence curves of SAPPSO-Rule, since robots constantly circumnavigate a concave obstacle but cannot avoid it, their convergence distances (d_c) show a wide range of fluctuations. From Fig. 13(e), the convergence curves of SAPPSO-Rule and SAPPSO-TS show that they converge to a small value. SAPPSO-TS enjoys faster convergence than SAPPSO-Rule since the latter needs to spend much time navigating around obstacles. In summary, TS shows better obstacle avoidance capability and adaptability to the environment with different obstacle's physical characteristics than its three peers.

TABLE VI
EXPERIMENTAL RESULTS IN SCENARIO 3

	SAPPSO-APF		SAPPSO-Fuzzy		SAPPSO-Rule		SAPPSO-TS	
	τ	s	τ	s	τ	s	τ	s
α=0, β=0.8	8.54	0.28	8.64	0.27	8.94	0.255	0	1
α=0.2, β=0.6	8.54	0.298	8.6	0.276	8.94	0.255	0	1
α=0.4, β=0.4	8.54	0.292	8.4	0.295	9.04	0.247	0	1
α=0.6, β=0.2	8.54	0.289	8.56	0.285	8.46	0.295	0	1
α=0.8, β=0	8.54	0.293	8.54	0.288	0.2	0.98	0	1

D. Discussion on Generating Tabu Areas

This work proposes a novel tabu search method to avoid concave obstacles for source location in an unknown environment, where a robot does not require any prior information about obstacles. As long as a robot enters a deadlock state, which prevents it from continuing to search, the proposed tabu search works to taboo deadlock-generating area. The deadlock state forms on two types of obstacle-related places, including inside and outside the concave obstacle, as shown in Fig. 14. From Fig. 14(a), $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are two robots outside and inside the concave obstacle. They are aiming to move towards the source Ω, but are trapped by the inner and outer walls of the concave obstacle. In Fig. 14(b), $\boldsymbol{x}_1$ generates tabu areas to escape the deadlock area. In Fig. 14(c), $\boldsymbol{x}_2$ generates tabu areas to fill up the concave obstacle. Finally, it is pushed out of the concave obstacle. From Fig. 14(d), $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ avoid the concave obstacle.

$$f_3 = f_3^* + \tilde{f}_3 \tag{18a}$$

$$f_3^*(x,y) = \begin{cases} 20 \cdot exp\left(-\left\|\dfrac{(x,y)-(0,-4)}{15}\right\|\right), x<0 \vee y<0 \vee y<\dfrac{2}{3}x-4 \\ 20 \cdot exp\left(-\left\|\dfrac{(x,y)-(0,-4)}{15}\right\|\right) + 20\alpha \cdot exp\left(-\left\|\dfrac{(x,y)-(0,4)}{15}\right\|\right), x>0 \wedge y<0 \wedge y<-\dfrac{2}{3}x+4 \\ 20\beta \cdot exp\left(-\left\|\dfrac{(x,y)-(0,-4)}{15}\right\|\right), x>0 \wedge y>0 \wedge y>\dfrac{2}{3}x-4 \end{cases} . \tag{18b}$$

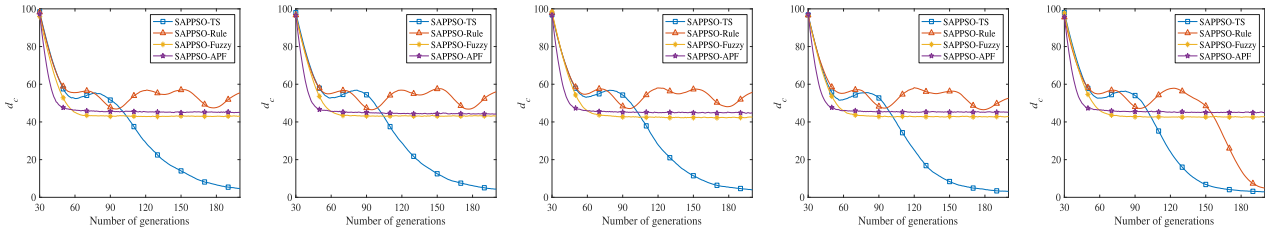
$$\tilde{f}_3(x,y) = \begin{cases} 12 \cdot exp\left(-\left\|\dfrac{(x,y)-(1,4)}{2}\right\|\right), (4<y<-8x+4) \vee (-\dfrac{4}{5}x+\dfrac{24}{5}<y<-\dfrac{4}{7}x+\dfrac{24}{7}) \\ 12\beta \cdot exp\left(-\left\|\dfrac{(x,y)-(1,4)}{2}\right\|\right), (y<-\dfrac{4}{5}x+\dfrac{24}{5} \wedge y<0) \vee (0<y<4 \wedge x<0) \\ g_1+g_2+g_3 \end{cases} \tag{18c}$$

$$g_1 = 12\alpha \cdot exp\left(-\left\|\frac{(x,y)-(-1,4)}{2}\right\|\right), (0<y<4 \wedge x>0) \vee -\frac{4}{7}x+\frac{24}{7}<y<0 \tag{18d}$$

$$g_2 = 12\alpha \cdot exp\left(-\left\|\frac{(x,y)-(1,-4)}{2}\right\|\right), (x>0 \wedge y>0 \wedge y>\frac{4}{5}x-\frac{24}{5}) \vee (x<0 \wedge y>-8x+4) \tag{18e}$$

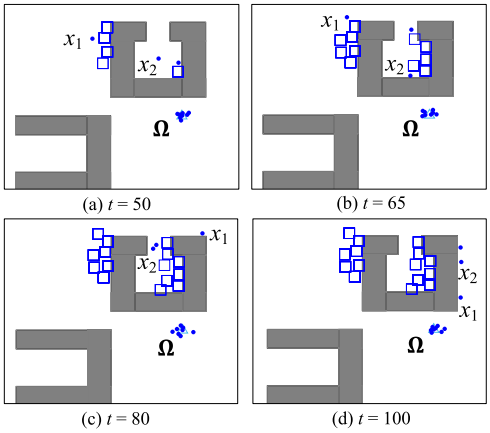
$$g_3 = 12 \cdot exp\left(-\left\|\frac{(x,y)-(1,4)}{2}\right\|\right), (x>0 \wedge y>0) \vee (x<0 \wedge y>-8x+4) \vee (y<0 \wedge y>-\frac{4}{7}x+\frac{24}{7}) \tag{18f}$$

Authorized licensed use limited to: TONGJI UNIVERSITY. Downloaded on January 27,2026 at 09:58:38 UTC from IEEE Xplore. Restrictions apply.



(a) α=0, β=0.8 (b) α=0.2, β=0.6 (c) α=0.4, β=0.4 (d) α=0.6, β=0.2 (e) α=0.8, β=0

Fig. 13. The convergence curves of robots in five signal diffusion conditions.



(a) t = 50 (b) t = 65 (c) t = 80 (d) t = 100

Fig. 14. The process of robot avoiding obstacles.

E. Discussion on the Use of Tabu Search in Real Scenarios

In real applications, each robot is equipped with a distance sensor, like a laser ranging sensor, to detect its distance from obstacles. The maximum sensing range of a robot is determined by the performance parameters of its equipped distance sensor. In this work, as long as the maximum sensing range of a distance sensor is greater than the maximum distance a robot moves in a time slice ($\widehat{V}$), it can be used to avoid concave obstacles.

The proposed PSO-TS uses a particle swarm optimizer for source location, where particles communicate with their neighbors to share their personal historical optimal position. Due to the constraints of the communication range, particle swarm optimizers with different topological structures are presented, including global topology [45], Euclidean distance-based local topology [46] and adaptive topology [47]. Then, in real-world scenarios, according to the different communication capabilities of robots, an adapted topology-based PSO can be used. How robots share information depends on their communication method, like global or local communication. If global communication is accessible, each robot stores all tabu areas. Robots will not repeatedly enter a concave obstacle that has been filled by tabu areas. If local communication is accessible, robots that can communicate with each other share their tabu areas. In this situation, a robot may re-enter a concave obstacle. Nevertheless, it can still avoid the concave obstacle by using the proposed tabu search method. Different communication methods only affect the efficiency of robots to find a source.

This work studies a source location problem in two-dimensional space, where a swarm of unmanned vehicles is used. The vehicle motion model mainly involves the control of a vehicle's pose (position coordinates, heading angle), speed, and front wheel rotation angle. A vehicle's motion model limits its instantaneous range of movement. In this work, before each time slice, PSO-TS is executed once to determine vehicles' next reachable positions. Algorithmically, PSO-TS does not restrict which motion model the robot uses to reach its next position. For simplicity, 360-degree rotatable robots are used in this work for simulation experiments.

VI. Conclusion

This work proposes a novel PSO based on Tabu Search (PSO-TS) for robots to locate multiple sources while avoiding concave obstacles. Instead of traditionally setting obstacles as tabu objects, it sets trapped areas as such objects. A weighted average velocity is computed for each robot to check if it is stuck inside concave obstacles. If so, the robot generates a tabu area, which can fill up a concave obstacle's open space. Then, robots are pushed out of the concave obstacle. For efficient search, an R-tree is adopted to cope with the storage of tabu objects. In the proposed method, robots need no prior information of unknown environments, thereby making the method adaptable to complex unknown environments. Experimental results show that the proposed tabu search has excellent algorithmic compatibility, environmental adaptability and obstacle avoidance capability. Its applications to various scenarios [48], [49], [50], [51], [52] should be pursued as our future work. Replacing PSO in this framework by other recently developed methods [53], [54] should be explores.

References

[1] R. Zou, V. Kalivarapu, E. Winer, J. Oliver, and S. Bhattacharya, "Particle swarm optimization-based source seeking," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 3, pp. 865–875, Jul. 2015.

[2] M. A. Demetriou, N. A. Gatsonis, and J. R. Court, "Coupled controls-computational fluids approach for the estimation of the concentration from a moving gaseous source in a 2-D domain with a Lyapunov-guided sensing aerial vehicle," *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 3, pp. 853–867, May 2014.

[3] P. Wu, F. Chu, A. Che, and M. Zhou, "Bi-objective scheduling of fire engines for fighting forest fires: New optimization approaches," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 4, pp. 1140–1151, Apr. 2018.

[4] L. Delobel, R. Aufrère, C. Debain, R. Chapuis, and T. Chateau, "A real-time map refinement method using a multi-sensor localization framework," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 5, pp. 1644–1658, May 2019.

[5] A. Sinha, R. Kumar, R. Kaur, and A. P. Bhondekar, "Consensus-based odor source localization by multiagent systems," *IEEE Trans. Cybern.*, vol. 49, no. 12, pp. 4450–4459, Dec. 2019.

[6] Z. Gao, J. Qin, S. Wang, and Y. Wang, "Boundary gap based reactive navigation in unknown environments," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 2, pp. 468–477, Feb. 2021.

Authorized licensed use limited to: TONGJI UNIVERSITY. Downloaded on January 27,2026 at 09:58:38 UTC from IEEE Xplore. Restrictions apply.

[7] H. Zhang, L. Jin, and C. Ye, "An RGB-D camera based visual positioning system for assistive navigation by a robotic navigation aid," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 8, pp. 1389–1400, Aug. 2021.

[8] S. Venkateswaran et al., "RF source-seeking by a micro aerial vehicle using rotation-based angle of arrival estimates," in *Proc. Amer. Control Conf.*, Jun. 2013, pp. 2581–2587.

[9] S.-J. Liu and M. Krstic, "Stochastic source seeking for nonholonomic unicycle," *Automatica*, vol. 46, no. 9, pp. 1443–1453, Sep. 2010.

[10] X. Jiang, S. Li, B. Luo, and Q. Meng, "Source exploration for an underactuated system: A control-theoretic paradigm," *IEEE Trans. Control Syst. Technol.*, vol. 28, no. 3, pp. 1100–1107, May 2020.

[11] K. Sakurama and S. Nishida, "Source seeking by distributed swarm robots with sample variance control," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2016, pp. 2484–2487.

[12] M. Cao, Q. Meng, Y. Jing, J. Wang, and M. Zeng, "Distributed sequential location estimation of a gas source via convex combination in WSNs," *IEEE Trans. Instrum. Meas.*, vol. 65, no. 6, pp. 1484–1494, Jun. 2016.

[13] P. Li, G. Pin, T. Parisini, and G. Fedele, "Deadbeat source localization from range-only measurements: A robust kernel-based approach," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2016, pp. 2729–2734.

[14] B. Chang, E. Chen, F. Zhu, Q. Liu, T. Xu, and Z. Wang, "Maximum *a posteriori* estimation for information source detection," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 50, no. 6, pp. 2242–2256, Jun. 2020.

[15] S. Al-Abri and F. Zhang, "A distributed active perception strategy for source seeking and level curve tracking," *IEEE Trans. Autom. Control*, vol. 67, no. 5, pp. 2459–2465, May 2022.

[16] S. Al-Abri, W. Wu, and F. Zhang, "A gradient-free three-dimensional source seeking strategy with robustness analysis," *IEEE Trans. Autom. Control*, vol. 64, no. 8, pp. 3439–3446, Aug. 2019.

[17] W. Wu and F. Zhang, "A speeding-up and slowing-down strategy for distributed source seeking with robustness analysis," *IEEE Trans. Control Netw. Syst.*, vol. 3, no. 3, pp. 231–240, Sep. 2016.

[18] G. Tian, Y. Ren, and M. Zhou, "Dual-objective scheduling of rescue vehicles to distinguish forest fires via differential evolution and particle swarm optimization combined algorithm," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 11, pp. 3009–3021, Nov. 2016.

[19] Q. Lu, Q. Han, and S. Liu, "A cooperative control framework for a collective decision on movement behaviors of particles," *IEEE Trans. Evol. Comput.*, vol. 20, no. 6, pp. 859–873, Dec. 2016.

[20] Q.-Y. Bao, S.-M. Li, W.-Y. Shang, and M.-J. An, "A fuzzy behavior-based architecture for mobile robot navigation in unknown environments," in *Proc. Int. Conf. Artif. Intell. Comput. Intell.*, vol. 2, Nov. 2009, pp. 257–261.

[21] M. F. Selekwa, D. D. Dunlap, D. Shi, and E. G. Collins, "Robot navigation in very cluttered environments by preference-based fuzzy behaviors," *Robot. Auto. Syst.*, vol. 56, no. 3, pp. 231–246, Mar. 2008.

[22] O. Motlagh, D. Nakhaeinia, S. H. Tang, B. Karasfi, and W. Khaksar, "Automatic navigation of mobile robots in unknown environments," *Neural Comput. Appl.*, vol. 24, nos. 7–8, pp. 1569–1581, Jun. 2014.

[23] R. Li and H. Wu, "Multi-robot source location of scalar fields by a novel swarm search mechanism with collision/obstacle avoidance," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 1, pp. 249–264, Jan. 2022.

[24] Q. Tang, F. Yu, Z. Xu, and P. Eberhard, "Swarm robots search for multiple targets," *IEEE Access*, vol. 8, pp. 92814–92826, 2020.

[25] Y. Wang, Y. Hu, J. Wang, and J. Tu, "Pollution source localization in complex environments using a swarm of mobile agents," *IEEE Sensors Lett.*, vol. 6, no. 1, pp. 1–4, Jan. 2022.

[26] Q. Tang, L. Ding, F. Yu, Y. Zhang, Y. Li, and H. Tu, "Swarm robots search for multiple targets based on an improved grouping strategy," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 15, no. 6, pp. 1943–1950, Nov. 2018.

[27] E. Masehian and M. R. Amin-Naseri, "Sensor-based robot motion planning—A Tabu search approach," *IEEE Robot. Autom. Mag.*, vol. 15, no. 2, pp. 48–57, Jun. 2008.

[28] X. Zhang, M. Yan, Y. Liu, and Y. Ju, "Autonomous navigation for mobile robot based on Tabu search in unknown environment," in *Proc. 29th Chin. Control Conf.*, Jul. 2010, pp. 3625–3630.

[29] L. Xiao Kang, L. Ming Yong, and Y. Mao De, "Tabu search based autonomous navigation algorithm for mobile robot," *Control Decis.*, vol. 26, no. 9, pp. 1310–1314, 2011.

[30] Z. Xin, Y. Maode, and J. Yongfeng, "A Tabu search based flocking algorithm of motion control for multiple mobile robots," in *Proc. 5th Int. Conf. Intell. Comput. Technol. Autom.*, Jan. 2012, pp. 48–52.

[31] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw. (ICNN)*, vol. 4, Aug. 1995, pp. 1942–1948.

[32] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. MHS 6th Int. Symp. Micro Mach. Hum. Sci.*, Oct. 1995, pp. 39–43.

[33] J. Tang, G. Liu, and Q. Pan, "A review on representative swarm intelligence algorithms for solving optimization problems: Applications and trends," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 10, pp. 1627–1643, Oct. 2021.

[34] Y. Cao, H. Zhang, W. Li, M. Zhou, Y. Zhang, and W. A. Chaovalitwongse, "Comprehensive learning particle swarm optimization algorithm with local search for multimodal functions," *IEEE Trans. Evol. Comput.*, vol. 23, no. 4, pp. 718–731, Aug. 2019.

[35] Y. Wang and X. Zuo, "An effective cloud workflow scheduling approach combining PSO and idle time slot-aware rules," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 5, pp. 1079–1094, May 2021.

[36] J. Bi, H. Yuan, S. Duanmu, M. Zhou, and A. Abusorrah, "Energy-optimized partial computation offloading in mobile-edge computing with genetic simulated-annealing-based particle swarm optimization," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3774–3785, Mar. 2021.

[37] D. Bratton and J. Kennedy, "Defining a standard for particle swarm optimization," in *Proc. IEEE Swarm Intell. Symp.*, Apr. 2007, pp. 120–127.

[38] F. Glover, "Tabu search: A tutorial," *Interfaces*, vol. 20, no. 4, pp. 74–94, Aug. 1990.

[39] O. A. Arık, "Population-based Tabu search with evolutionary strategies for permutation flow shop scheduling problems under effects of position-dependent learning and linear deterioration," *Soft Comput.*, vol. 25, pp. 1–18, Aug. 2020.

[40] G. Garai and B. B. Chaudhurii, "A novel hybrid genetic algorithm with Tabu search for optimizing multi-dimensional functions and point pattern recognition," *Inf. Sci.*, vol. 221, pp. 28–48, Feb. 2013.

[41] R. Bayer and E. McCreight, "Organization and maintenance of large ordered indexes," in *Proc. Rec. ACM SIGFIDET Workshop Data Description Access*. Houston, TX, USA: Rice Univ., Nov. 1970, pp. 107–141.

[42] A. Guttman, "R-trees: A dynamic index structure for spatial searching," *SIGMOD Rec.*, vol. 14, no. 2, pp. 47–57, Jun. 1984.

[43] J. Li and Y. Tan, "A probabilistic finite state machine based strategy for multi-target search using swarm robotics," *Appl. Soft Comput.*, vol. 77, pp. 467–483, Apr. 2019.

[44] Z. Zheng, J. Li, J. Li, and Y. Tan, "Improved group explosion strategy for searching multiple targets using swarm robotics," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2014, pp. 246–251.

[45] J. Li, J. Zhang, C. Jiang, and M. Zhou, "Composite particle swarm optimizer with historical memory for function optimization," *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2350–2363, Oct. 2015.

[46] B. Y. Qu, P. N. Suganthan, and S. Das, "A distance-based locally informed particle swarm model for multimodal optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 3, pp. 387–402, Jun. 2013.

[47] N. Zeng, Z. Wang, W. Liu, H. Zhang, K. Hone, and X. Liu, "A dynamic neighborhood-based switching particle swarm optimization algorithm," *IEEE Trans. Cybern.*, vol. 52, no. 9, pp. 9290–9301, Sep. 2022.

[48] L. Meng, Q. Kang, C. Han, and M. Zhou, "Determining the optimal location of terror response facilities under the risk of disruption," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 2, pp. 476–486, Feb. 2018.

[49] L. Huang, M. Zhou, K. Hao, and E. Hou, "A survey of multi-robot regular and adversarial patrolling," *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 4, pp. 894–903, Jul. 2019.

[50] L. Huang, M. Zhou, and K. Hao, "Non-dominated immune-endocrine short feedback algorithm for multi-robot maritime patrolling," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 1, pp. 362–373, Jan. 2020.

[51] P. Wu, F. Chu, N. Saidani, H. Chen, and M. Zhou, "Optimizing locations and qualities of multiple facilities with competition via intelligent search," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 5092–5105, Jun. 2022.

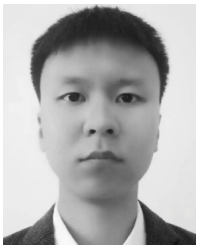
[52] Z. Xing, X. Chen, X. Wang, W. Wu, and R. Hu, "Collision and deadlock avoidance in multi-robot systems based on glued nodes," *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 7, pp. 1327–1330, Jul. 2022.

[53] M. Cui, L. Li, M. Zhou, and A. Abusorrah, "Surrogate-assisted autoencoder-embedded evolutionary optimization algorithm to solve high-dimensional expensive problems," *IEEE Trans. Evol. Comput.*, vol. 26, no. 4, pp. 676–689, Aug. 2022.

[54] Y. Zhou, G. Wang, and M. Zhou, "Detecting k-vertex cuts in sparse networks via a fast local search approach," *IEEE Trans. Computat. Social Syst.*, 2023, doi: 10.1109/TCSS.2023.3238042.

Authorized licensed use limited to: TONGJI UNIVERSITY. Downloaded on January 27,2026 at 09:58:38 UTC from IEEE Xplore. Restrictions apply.

Junqi Zhang (Senior Member, IEEE) received the Ph.D. degree in computing science from Fudan University, Shanghai, China, in 2007. In 2007, he was a Post-Doctoral Research Fellow with Peking University, Beijing, China. From 2014 to 2015, he was a Visiting Scholar with the New Jersey Institute of Technology, Newark, NJ, USA. He is currently a Full Professor with Tongji University, Shanghai. He has published more than ten papers in IEEE TRANSACTIONS and more than 30 papers in conferences. His current research interests include swarm intelligence, swarm robots, multi-agent systems, learning automata, reinforcement learning, and big data. He was a recipient of the Outstanding Post-Doctoral Award from Peking University.



Huan Liu received the M.S. degree in computer science and technology from the Lanzhou University of Technology, Lanzhou, China, in 2018. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Technology, Tongji University, Shanghai, China. His current research interests include evolutionary computation and its applications.

Peng Zu received the B.S. degree in computer science and technology from Tongji University, Shanghai, China, in 2021, where he is currently pursuing the M.S. degree with the Department of Computer Science and Technology. His current research interests include swarm robots, multi-agent systems, and learning automaton and its applications.

Mengshi Zhao is currently pursuing the B.Sc. degree with Tongji University, Shanghai, China. His research interests include computation intelligence and optimization theory.

Cheng Wang (Senior Member, IEEE) received the M.S. degree from the Department of Applied Mathematics, Tongji University, in 2006, and the Ph.D. degree from the Department of Computer Science, Tongji University, in 2011. He is currently a Professor with the Department of Computer Science, Tongji University. His research interests include cyberspace security and intelligent information services.

Aiiad Albeshri received the M.S. and Ph.D. degrees in information technology from the Queensland University of Technology, Brisbane, QLD, Australia, in 2007 and 2013, respectively. Since 2018, he has been an Associate Professor with the Department of Computer Science, King Abdulaziz University, Jeddah, Saudi Arabia. His current research interests include information security, trust in cloud computing, big data, and HPC.

Abdullah Abusorrah (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the University of Nottingham, U.K., in 2007. He is currently a Professor with the Department of Electrical and Computer Engineering, King Abdulaziz University. He is also the Head of the Center for Renewable Energy and Power Systems, King Abdulaziz University. His research interests include energy systems, smart grids, and system analysis.

MengChu Zhou (Fellow, IEEE) received B.S. degree in control engineering from the Nanjing University of Science and Technology, Nanjing, China, in 1983, the M.S. degree in automatic control from the Beijing Institute of Technology, Beijing, China, in 1986, and the Ph.D. degree in computer and systems engineering from Rensselaer Polytechnic Institute, Troy, NY, USA, in 1990. He joined the Department of Electrical and Computer Engineering, New Jersey Institute of Technology in 1990, and is now a Distinguished Professor. He has over 1100 publications including 14 books, over 750 journal papers including over 600 IEEE Transactions papers, 31 patents and 32 book-chapters. His interests are in intelligent automation, robotics, Petri nets, Internet of Things, edge/cloud computing, and big data analytics. He is a life member of Chinese Association for Science and Technology-USA and served as its President in 1999. He is Fellow of International Federation of Automatic Control (IFAC), American Association for the Advancement of Science (AAAS), Chinese Association of Automation (CAA) and National Academy of Inventors (NAI). He is a recipient of Excellence in Research Prize and Medal from NJIT, Humboldt Research Award for US Senior Scientists from Alexander von Humboldt Foundation, and Franklin V. Taylor Memorial Award and the Norbert Wiener Award from IEEE Systems, Man, and Cybernetics Society, and Edison Patent Award from the Research & Development Council of New Jersey.

Authorized licensed use limited to: TONGJI UNIVERSITY. Downloaded on January 27,2026 at 09:58:38 UTC from IEEE Xplore. Restrictions apply.